The 2017 Conference on Computational Linguistics and Speech Processing
ROCLING 2017, pp. 53-67

應用興趣點辨識技術從 Web 中挖掘新商家資訊

Mining POIs from Web via POI recognition and Relation Verification

許國信　Kuo-Hsin Hsu
國立中央大學資訊工程學系
Department of Computer Science & Information Engineering
National Central University
105522092@cc.ncu.edu.tw

莊秀敏　Hsiu-Min Chuang
國立中央大學資訊工程學系
Department of Computer Science & Information Engineering
National Central University
showmin1205@gmail.com

周建龍　Chien-Lung Chou
國立中央大學資訊工程學系
Department of Computer Science & Information Engineering
National Central University
formatc.chou@gmail.com

張嘉惠　Chia-Hui Chang
國立中央大學資訊工程學系
Department of Computer Science & Information Engineering
National Central University
chia@csie.ncu.edu.tw

摘要

本論文提出一套系統能從網頁中自動化的挖掘新的店家資訊的方法。透過地址相關的特殊的關鍵字(如：台北市+新開幕)進行搜尋，找到可能包含地址及新開幕店家的網頁，再利用地址辨識模型先從結果中擷取地址，並從周圍透過興趣點辨識模型擷取商家名稱(Store Name Recognition)，最終使用地址與興趣點關聯配對(POI Relation)模型推斷該商家名稱是否位於該地址。我們特別著重在商家名稱辨識以及 POI Relation 的模型建立。針對興趣點辨識模型的資料準備，我們將黃頁上的商家名稱透過實體篩選以及資料前處理，應用 Distant Learning 及序列標記，可以訓練出 F1 值 0.816 的興趣點辨識模型。其次關於 POI Relation 預測則是針對反例的準備進行研究，其中效能最好的模型有 0.754 的準確率。整體系統效能則使用兩個興趣點辨識模型搭配三種關聯分類模型，共進行六次實驗並分析，最好的組合平均每個 IP 每天能找到約 49 個新的興趣點。

Abstract

This paper presents a system that could automatically extract new POIs from Web. First, we use special queries (e.g. Taipei+New Open) to find Web pages that might contain addresses for new stores. For web pages that contain addresses, we then apply store name recognition model to extract possible POIs. Finally, we train a model to find the most possible POI for the address found in the page. In this paper, we focus on POI name recognition and POI relation prediction. For POI recognition, we use store names from yellow pages as seed to prepare the training data via distant learning. Through entity selection and data processing, we obtain a model with 0.816 F1-measure as opposed to 0.432 F1-measure for a dictionary-based baseline. As for POI relation prediction, we compare three different strategies for negative example preparation. The best model could get 0.754 accuracy. We combine two POI recognition models with three classification models to test the overall performance. The best combination could extract 49 POIs every day with a single IP.

關鍵詞：興趣點辨識模型、二元分類關聯分類模型

Keywords: Address Recognition, POI Entity Recognition, POI Relation Prediction

一、緒論

隨著無線網路和智慧型手機的普及，傳統翻閱電話簿或名片的方式大幅減少，使用者們開始習慣利用網路查詢店家資訊。因此，谷歌、雅虎、微軟和諾基亞等公司很早就已開始開發商業地圖以滿足這類的需求，也有其他公共的地圖，像是：OpenStreetMap 和 OpenPOI，用於建設和維護興趣點(Point of Interest, POI)數據庫。

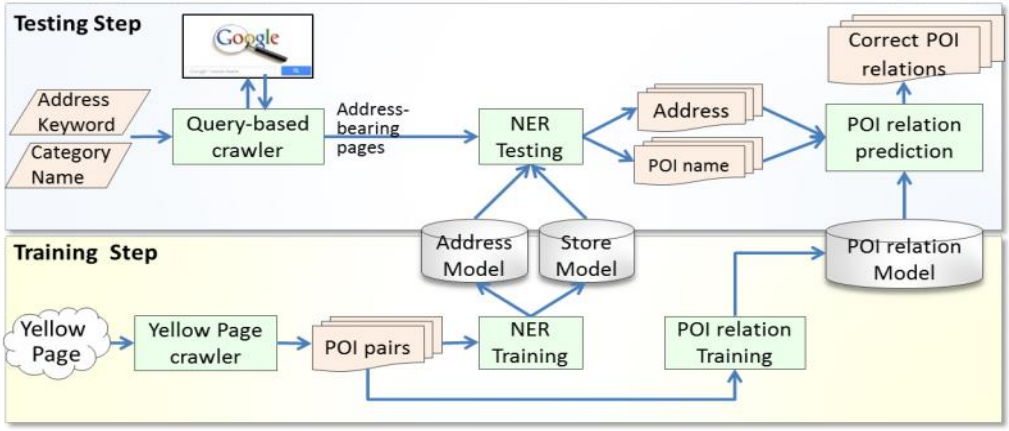
在 W3C 的定義中，興趣點可以視為一個擁有可用信息的位置。廣義上來說，任何能在電子地圖上標記的某個地標或是建築，都可以當作興趣點，像是：中央大學、捷運中山站、台北 101 等等。

不同的地圖服務可能有各自的特色，但多數都會提供興趣點的地址，若興趣點為店家或是機構則多會增加電話、營業時間等資訊。而地圖服務著重在資訊的正確性上，若提供的資訊與事實不符，將會造成使用者困擾，容易因為錯誤資訊而被給予較低分的評價，導致口碑不佳無人使用的窘境。

地圖服務的問題是如何蒐集新的資料。根據財政部的統計，我國餐飲業之營利事業家數

在民國 103 年共有 117,307 家，民國 104 年則增加到 124,124 家，平均每天增加 18 家。地圖服務不僅僅包含餐飲店家，尚有旅遊、醫療、學校、五金、水電等其他店家，平均每天新增的興趣點可能有上百個。不可能只倚靠真人考察或瀏覽部落客文章去新增興趣點，因此需要自動化。

由於我們可以利用地址關鍵字(區、市、鎮)做為搜尋字去抓取網頁中的地址，因此本論文以地址的角度去擬定策略，希望快速地從網路上挖掘出新的地址，並給予每個新的地址正確的興趣點名稱，以此自動化擴充資料庫。我們提出的系統包含四個部分：第一部分為關鍵字爬蟲，第二部分為地址辨識，第三部分為興趣點辨識，最後則是配對關係預測，如圖一所示。系統可以透過特定的關鍵字(ex:台北市+新開幕)進行第一次資料蒐集，從中擷取出辨識的地址，以及包含地址網頁中的興趣點，最後對每個地址和其找到的興趣點做關係預測，選擇機率最高的做為正確配對。



圖一、系統架構圖

雖然將 Google 搜尋引擎作為大量資訊的來源，抓取該關鍵字的前十篇搜尋結果可以快速的找尋新的商家，不過這項搜尋來源的限制是同 IP 不能頻繁地向 Google 搜尋引擎蒐集資料，因此系統每天能自動化找到多少個新的商家是實作地理資訊系統所關心的主題。本篇論採用 Huang [12]之方法擷取中文地址(F1 值可達 97.2%)，並改善興趣點辨識達到 81.6%的 F1 值，另外地址與商家配對驗證模型精準率為 74.56%，系統每天能自動化找到 49 個新的興趣點。

本論文的內容組織如下：第二章描述相關研究。第三章為系統架構及 POI 名稱辨識，第四章為 POI 與地址關係預測模型，第五章為實驗數據以及實驗結果，最後第六章提出結論和未來研究方向。

二、相關研究

實體提取是從非結構化文本文檔中識別命名實體的任務，這是用於測試機器能夠理解自然語言寫入的消息以及自動執行通常執行的常規任務的信息任務之一。現今常用的方式是以序列標記實體的開始、中繼、結束、其他作為擷取的參照，並以隱藏馬爾可夫模型（Hidden Markov Model, HMM）和條件隨機場（Conditional Random Field, CRF）為主要技術[11]。由於監督學習需準備大量的訓練資料，而人工標記需要相關知識且耗費時間。因此 Chou[1]與 Huang[12]等人即提出利用已知的實體清單進行自動標記進而生成訓練資料的 Distant Learning 架構。本論文中的興趣點辨識模型即仿效黃的做法，並改善辨識效能，此為本研究的第一個主題。

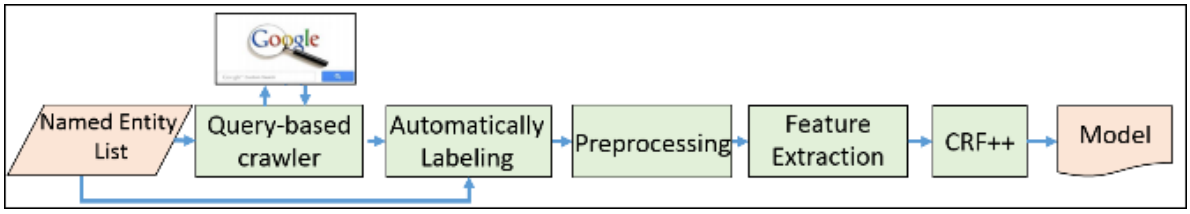
本文第二個主題則是地址與興趣點關係的驗證模型。提取實體之間的語義關係是數據鏈接與本體建構發展的關鍵步驟。大多數的研究著重在二元關係的擷取[3]，某些監督式學習的研究提出特徵導向[4]以及核心(kernel-based)導向[5]的方法。由於監督式學習需要大量標記資料，因此半監督式學習(Semi-supervised Learning)和自助法(bootstrapping)就顯得重要，DIPRE [6]和 Snowball [9]分別使用一小組標記種子實例和手工提取格式來訓練模型。而 KnowItAll[8]和 TextRunner[7]則是採用自我訓練之大型關係擷取系統。

本文採用高霆耀及莊秀敏[13]等人的作法，透過搜尋地址與店家的結果數、皮爾森相關係數 (Pearson correlation coefficient)、餘弦相似度(Cosine similarity)等共 27 個特徵去推斷該商家是否位在該地址上，並藉由黃頁的商家資訊準備正反配對訓練及測試資料，然實務上地址與商家配對之測試資料與訓練資料並不相同，為加速系統運作，本文提出新的訓練資料準備方式，希望可以提升系統運作效能。

三、興趣點辨識模組

興趣點辨識模組包含五個步驟，包括以已知興趣點作為關鍵字查詢可能包含興趣點的句子、自動標記、資料前處裡、特徵擷取以及使用 CRF++進行模型訓練，如圖二所示。我們從中華黃頁上搜集 677,172 個商家興趣點做為實體清單，然而一開始所得到的模型效果並不佳，原因是黃頁中商家名稱可能使用註冊人名、食物名稱、類別名稱(土木工程)或地區名稱(桃園市中壢區、高樹鄉)做為興趣點，導致所得模型標記準確率太低，因

此需要進一步的篩選，過濾掉不符合興趣點定義的實體。



圖二、興趣點辨識模組流程圖

3.1 商家實體篩選

由於黃頁中的興趣點可能包含英文、數字或特殊符號，而我們的研究著重在純中文的興趣點名稱，因此保留由中文以及括號組成的實體，其餘全部去除。保留括號是因為某些興趣點會跟隨著區域名，像是連鎖店就會透過不同門市來區分，舉例來說「全家(中央店)」和「全家(中正店)」同樣都是便利超商，但由於落在不同區域也就會有不同的門市名稱。

篩選的規則可以依據實體的長短分成兩部分(長度為五到十五為長實體；長度三或四則是短實體)，其中針對短實體可能為人名的部分再加以細分兩種規則。之後使用 1,563 個食物名稱以及 1,275 個類別名稱進行過濾，並利用正規表達式(區鄉鎮縣市部)去除地區名稱，避免擷取出大範圍的地點名稱，留下長實體的興趣點，以下簡稱此清單為 L。

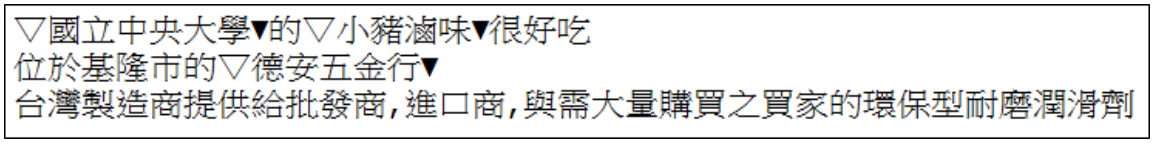
接著，我們透過正規表達式以及人名辨識模型過濾註冊人名。正規表達式的方法係利用 124 個常用姓氏做為開頭，獲得去除可能人名後的短實體興趣點，以下簡稱此清單 SR。以人名辨識模型的部分，則先以短實體做為搜尋關鍵字爬取前十篇搜尋結果，並利用人名辨識模型擷取人名清單[12]，和原先的短實體清單取差集後即為過濾完成的短實體清單，以下簡稱此清單 SP。

我們將 L 和 SR 聯集後取得 556,036 個實體，透過搜尋為每個實體取得相關句子做為訓練資料，經處理後共有 928,567 句。而利用 L 和 SP 聯集後則有 556,702 個實體，以及 916,383 訓練句。而原先 677,172 個黃頁興趣點，則有 1,560,622 包含實體的句子。

3.2 自動標記

採用已知實體名稱，自動標記句子生成訓練資料，是解決人工標記成本過於昂貴的方法。

雖然句子是由查詢已知興趣點取得，但由於每句話可能包含一個或一個以上的興趣點，也可能完全沒有（如圖三）。當句子數量與實體名稱數量均大於幾十萬時，自動標記的成本就會相當大。為避免巢狀標記(即一個 POI 裡包含另一個 POI)與加速標記速度，我們將興趣點依長度由大至小排列，並將比對成功的部份去除後，再比對較短的實體。舉例而言，「國立中央大學的小豬滷味很好吃」比對到「國立中央大學」之後，句子縮減成「的小豬滷味很好吃」，再與剩餘的興趣點比對，若是縮減後的句子長度小於 2，則可直接結束比對流程。



圖三、自動標記範例圖（▽和▼符號代表實體的開始以及結束）

3.3 特徵擷取

資料來源五花八門，可能來自新聞、部落格文章或是社群的貼文，每個句子闡述的內容不盡相同，因此我們利用興趣點前後的字詞做為特徵，再用 CRF++進行模型的訓練。此處使用逗號以及句號做為斷句的依據，在資料量最大的實驗中，斷句後有 5,323,009 句，總字數更是破億字，基於設備以及訓練速度的考量，我們只保留含有興趣點的句子，也就是去除所有的不包含興趣點句子的負範例(negative example)。經過實驗後發現效能沒有明顯下降，實驗時間則有大幅度的降低。

我們依據 Chou[1]提出的五類十四種特徵(如表一)：是否為實體前常出現的詞(Common Before)、是否為實體後常出現的詞(Common After)、常出現的實體前綴詞是(Common Prefix)、常出現的實體後綴詞(Common Postfix)，以及是否為特殊(如英文、數字等)符號。除了第十三和第十四個特徵有固定的字典，剩餘的字典會從訓練資料中擷取，並透過該字出現的頻率進行篩選。

最後我們利用 CRF++進行訓練，將序列標記成 BIEO 符號，B 代表實體的開始，I 代表實體的中間字元，E 代表實體的結束，O 則是代表不屬於實體。

表一、興趣點辨識模型特徵表

ID	Name	Description
1	Before_1	unigram word before entity or not
2	Before_2	bigram word before entity or not
3	Before_3	trigram word before entity or not
4	Prefix_1	prefix unigram word or not
5	Prefix_2	prefix bigram word or not
6	Prefix_3	prefix trigram word or not
7	Sufffix_1	suffix unigram word or not
8	Sufffix_2	suffix bigram word or not
9	Sufffix_3	suffix trigram word or not
10	After_1	unigram word after entity or not
11	After_2	bigram word after entity or not
12	After_3	trigram word after entity or not
13	English/Number	English or number?
14	Symbol	Symbol or not?

四、地址與興趣點關聯分類

如緒論所述，本論文首先應用地址關鍵字對搜尋引擎查詢得到可能包含地址的網頁，再從擷取出新的地址網頁中辨識可能的興趣點名稱，最後進行配對關係預測（如圖一所示）。本節的目的即在判斷給定地址 a 與興趣點 p 之間的配對關係。基本上系統將對搜尋引擎分別送出三個查詢：a、p、及 a+p 以得到搜尋結果：T_a 代表地址的前十篇搜尋結果，T_p 代表興趣點的前十篇搜尋結果，T_{a+p} 代表地址與興趣點的前十篇搜尋結果。為了有效地辨識出地址和興趣點關聯，我們定義以下如表二、共十二個特徵。

表二、地址與興趣點關聯分類特徵表

ID	Name	Query a	Query p	Query a+p	Description
1	C(a)	●	○	○	# of search results for query a in normalized scale
2	C(p)	○	●	○	# of search results for query p in normalized scale
3	C(a, p)	○	○	●	# of search results for query a+p in normalized scale
4	R(a+p|a)	●	○	●	the ratio of C(a+p) to C(a)
5	R(a+p|p)	○	●	●	the ratio of C(a+p) to C(p)
6	P(a+p|T_a)	●	○	○	the percentage of top 10 snippets from T_a that support the POI relation (a,p)
7	P(a+p|T_p)	○	●	○	the percentage of top 10 snippets from T_p that support the POI relation (a,p)
8	P(a+p|T_{a+p})	○	○	●	the percentage of top 10 snippets from T_{a+p} that support the POI relation (a,p)
9	NDCG(p|T_a)	●	○	○	the rank of p in top 10 snippets from T_a
10	NDCG(a|T_p)	○	●	○	the rank of a in top 10 snippets from T_p
11	cos(T_a, T_p)	●	●	○	the cosine similarity for snippet T_a and T_p
12	D(a+p)	○	○	●	Today – D(a+p) in log scale

特徵一到特徵三利用搜尋結果數取對數而得。我們認為地址或興趣點的搜尋結果數越低，代表其不存在的機率越高，在驗證時被分類為錯誤配對(False)的機率就會越高。以地址與興趣點作為關鍵字的搜尋結果數越多，代表地址與興趣點間的關聯性越高。特徵四和特徵五透過計算條件機率取得，和特徵三有著相同性質，數值越大代表關聯性越大。

特徵六到特徵八採用 co-occurrence 的方法計算，利用地址、興趣點或是地址與興趣點作為關鍵字的搜尋結果，計算地址和興趣點同時出現的機率，該特徵值越大代表兩者一起提到的機率越大，關聯性也就愈高。

若該地址出現在該興趣點的搜尋結果之第一篇，代表兩者的關聯越高；反之若在最後一篇才有提及、甚至沒有出現，則代表關聯較低。因此我們採用 NDCG 作為第九和第十個特徵。NDCG 是種計算排名的方法，常用來測量搜尋引擎的演算法是否有效。

第十一個特徵是餘弦相似度，數值越高代表兩者的相似度越高，關聯性也就越高。最後一個特徵和時間有關，我們認為越新的資訊越正確。因此，利用正規表達式從搜尋結果中辨識時間，減系統時間後取對數得到此特徵值，該值越大表示越舊；反之該值越小表示資料越新，正確性亦會較高。

五、實驗

本節內容包含三個部份，分別是興趣點辨識效能的比較、地址與商家關係預測、以及整體系統效能。

5.1 興趣點辨識

興趣點辨識評估方式採取部分比對，假如標準答案是「蔣中正紀念館」，而模型只辨識出「紀念館」，會得到 0.5 分，部分比對分數、精準率(Precision)、召回率(Recall)以及 F1 值之算法如以下。

$$Score_p = \frac{|Overlap\ tokens|}{|Identified\ entity\ tokens|} \quad Score_r = \frac{|Overlap\ tokens|}{|Real\ entity\ tokens|}$$

$$Precision = \frac{\sum Score_p}{|Identified\ entities|} \quad Recall = \frac{\sum Score_r}{|Real\ entities|}$$

$$F - Measure = \frac{2PR}{P+R}$$

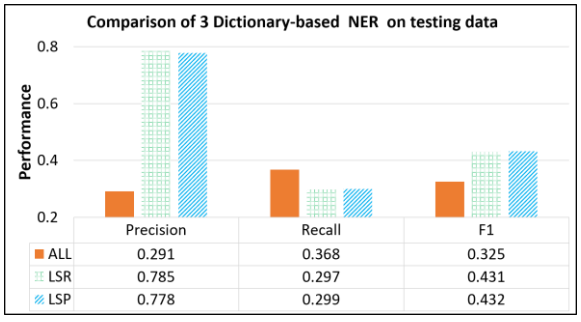
表三、測試資料之一致性信度值

		Labeler1				
		B	I	E	O	SumL2
Labler2	B	5,161	28	0	621	5,810
	I	60	24,483	232	3,204	27,979
	E	0	80	4,937	793	5,810
	O	583	2,025	635	367,859	371,102
	SumL1	5,804	26,616	5,804	372,477	410,701

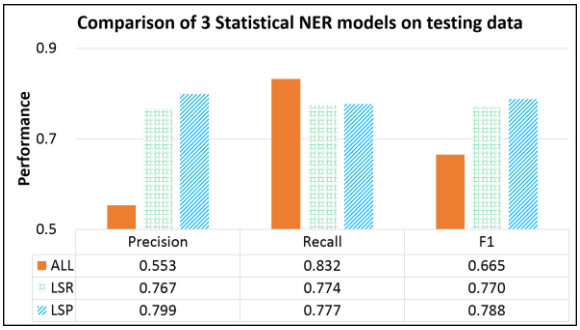
興趣點辨識的測試資料採用人工標記，以 250 個類別名稱、1,000 個食物名稱、1,000 個地址以及 1,000 個興趣點做為搜尋關鍵字，共爬取 4,000 個搜尋結果。先以自動標記的方法對訓練資料進行答案標記，再請兩位標記人員修正錯誤以及補標答案。標記之一致性信度(Kappa)值為 0.886，表示標記答案的可信度，如表三**錯誤! 找不到參照來源。**所示。

5.1.1 興趣點篩選效能分析

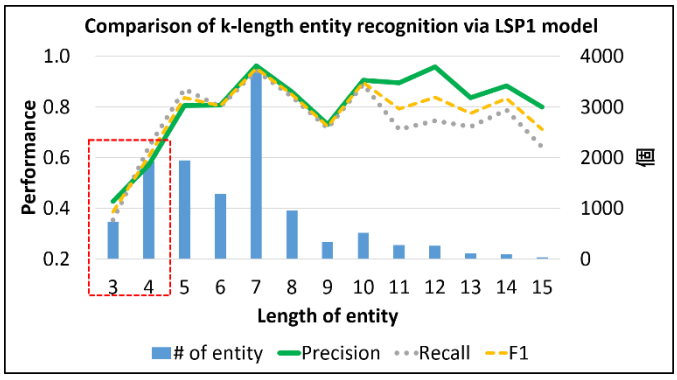
我們首先比較使用原始黃頁商家(ALL)，與篩選過的長實體 L 聯集兩種短實體 SR 及 SP 作為已知興趣點利用自動標記所得的辨識效能（圖四），並與自動標記所訓練出的模型（圖五、三種實體模型辨識效能比較圖）做比較。從圖四中可以看出，僅用字典比對方法的效能有限，沒有篩選過的原始黃頁商家僅有 0.291 的準確率，雖然篩選過人名的準確率可達 0.785 及 0.778，但是召回率不到 0.3；其中使用 LSR 清單的最佳效能，F1 值也只有 0.432。而經過訓練的模型，即使是沒有經過篩選(ALL)的模型 F1 效能也有 0.665；表現最好的是過濾人名的 LSP 辨識模型，F1 效能為 0.788，而正規表達式人名過濾所訓練的模型效能則達到 0.770。



圖四、字典導向之興趣點辨識效能



圖五、三種實體模型辨識效能比較圖

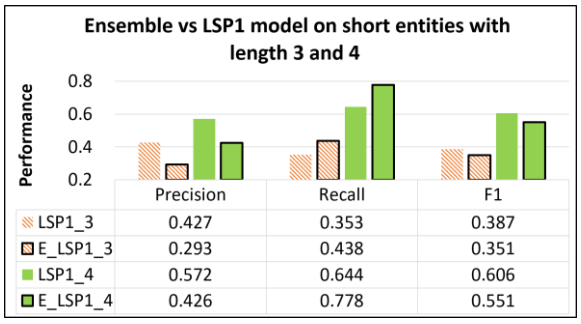


圖六、LSP1 模型之不同長度實體效能圖

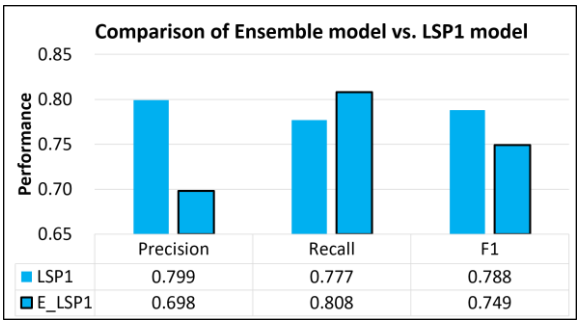
5.1.2 短興趣點效能提升

為改善興趣點辨識模型效能，我們分別針對不同長度的實體進行分析，以先前提及的清單 L 和清單 SP 爬取第一筆搜尋結果命名為 LSP1，如圖六所示。可以看到長度三和四的短實體的效能低於整體效能，因此我們嘗試兩種方法去提升短實體辨識效能：合併模型以及增加短實體資料量。

我們利用先前提及的清單 L 和清單 SP 爬取第一筆搜尋結果(LSP1)，並分別訓練模型，再聯集兩個模型的答案，做為最終的標記結果。其中 L 共有 491,330 個實體、773,927 句；SP 有 65,372 個實體、108,874 句。從圖七可以看出，合併模型(E_LSP1)在長度三、四實體以及整體的精準率皆低於一般模型(LSP1)，即使召回率都比一般模型來的好，整體看來 F1 值都較低，因此合併模型的方式無法有效改善效能。

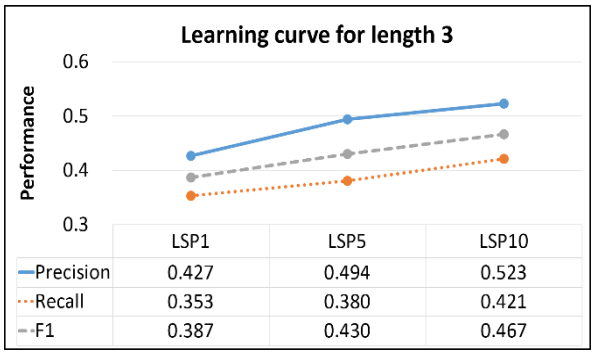


圖七、合併模型短實體部分之效能比較圖

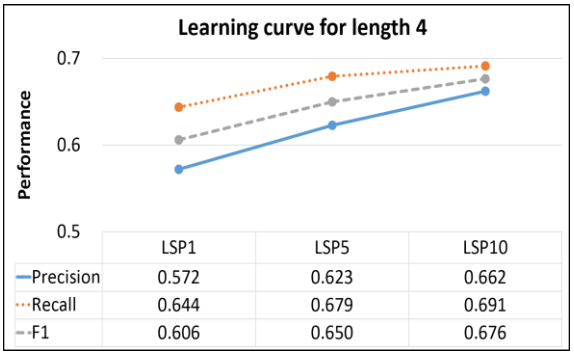


圖八、合併模型整體之效能比較圖

第二種做法則是將短實體的搜尋結果從一筆增加至五筆和十筆。圖九以及圖十分別顯示長度 3 及 4 的實體辨識效能，我們發現增加訓練資料量可以有效提升效能，F1 值從原先的 0.387 提升至 0.467。長度為四的興趣點辨識則是從 0.606 提升至 0.676。

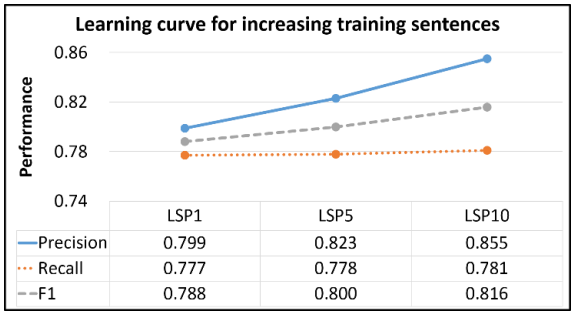


圖九、增加短實體訓練資料量對長度為 3 興趣點之學習曲線圖



圖十、增加短實體訓練資料量對長度為 4 興趣點之學習曲線圖

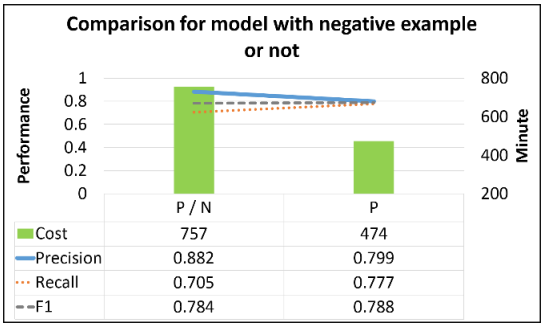
從整體效能的角度來看，增加短興趣點資料量可使精準率從 0.799 提升至 0.855(如圖十一)；召回率雖沒有明顯成長，也從 0.777 來到 0.781；F1 值則是從 0.788 改善至 0.816。



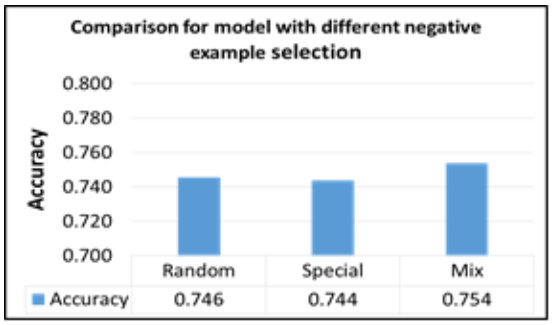
圖十一、增加短興趣點訓練資料之整體效能比較圖

5.1.3 訓練模型效率之提升

在效率上，比較保留所有搜尋結果的全部句子 (2,426,201 句)所訓練的模型，與去除不包含興趣點的句子(916,383 句)所訓練的模型。從圖十二可看出，去除不包含興趣點的模型和保留的模型效能相差不遠，實驗時間卻從 757 分鐘降到 474 分鐘，共減少 37.3%的時間。由此可知，去除不包含興趣點能大幅降低訓練時間，且效能不會有太多的變動。



圖十二、去除不包含興趣點之效能及時間比較圖



圖十三、地址與興趣點關聯模型效能比較圖

5.2 地址與興趣點關聯預測

實驗的第二部份則是 POI Relation 的預測。我們從黃頁上搜集並篩選 4,000 出個正確的地址與興趣點配對做為正例(Positive example)，而反例(Negative example)的挑選分成三種：從正例中隨機挑選興趣點去和地址做配對，若配對結果為正例則去除並重新隨機挑選，直到反例數量達到 4,000 個。第二種則是模仿系統真實運作時從地址的搜尋結果中辨識到其他興趣點並與其地址配對成的反例；最後則是綜合兩種方法做為第三種不同的準備方式。測試資料則挑選和訓練資料不同的 2,500 個正例，使用第一種和第二種方式各準備 1,250 個反例並進行人工標記，最終資料共包含 2,740 個正例以及 2,560 個反例。

從圖十三中可以看出三個模型的效能差不多，其中第二種的反例準備方式效能最低，準確率為 0.744，但是混合訓練資料的效果則得到最高的 0.754 準確率。問題可能出在訓練資料的準備中，辨識出的興趣點或許的確落在該地址上，然而該配對並不存在黃頁中的資料中，因此被我們做為反例進而降低效能。

5.3 系統效能測試

最後我們利用兩個興趣點辨識模型(ALL, LSP10)以及三個關聯模型進行共六次的系統測試。由於一個地址附近可能會有提到數個興趣點(表四第 2 行)，我們保留驗證機率大於 0.5 的候選興趣點(表四第 3 行)並進行排序，選出最高機率的候選者做為該地址配對的興趣點，最終人工標記結果並計算準確率(表四第 4 行)。其中利用 ALL 模型辨識出的興趣點共有 816 個，LSP10 則辨識出 340 個興趣點，因此前三組所需要的驗證時間也是後者的 2.4 倍，再由此花費的時間推算每日可以找到新（正確）的地址商家配對。

從表四中可以看出，利用 LSP10 搭配第一種關聯分類模型的效能最好，準確率達到 0.648，總花費時間為 1,034 分鐘，預估每個 IP 每天能找到約 49 個新的興趣點。而 ALL 模型搭配第一種關聯分類模型的效能只有 0.291，則是因為所辨識出的興趣點不正確導致效能以及效率降低。

不論是利用第二種或第三種關聯分類模型的組合之效率都極差，可能是因為訓練資料中有正確配對被視為反例，導致實際測試時正確配對不能被成功分類，從候選人的數量中

即可看出，利用隨機分配產生反例的關聯分類模型能找到較多的 POI，數量為其他兩個的幾十倍。這背後的原因可能是因為正例和反例的差異較大，能輕易將正確答案與錯誤答案分類，而第二種和第三種方法仿效真實系統運作狀況，包含和正例較相近的反例，舉例來說「全家(中央店)」是正確答案，而「全家」出現在反例中，即有可能影響結果。

表四、系統測試結果比較表(Efficiency for correct POI = Accuracy * Efficiency)

	# of POI	# of candidate	Accuracy	Cost (min)	Efficiency (POI/day)	Efficiency for correct POI (POI/day)
ALL + Random	816	277	0.291	2,307	53.70	15.63
ALL + Special	816	13	0.182	2,311	1.25	0.23
ALL + Mix	816	3	1	2,306	1.25	1.25
LSP10 + Random	340	88	0.648	1,034	75.20	48.73
LSP10 + Special	340	2	1	1,033	1.39	1.39
LSP10 + Mix	340	2	1	1,034	1.39	1.39

六、結論

我們以黃頁商家興趣點做為已知實體名稱，應用搜尋引擎收集包含興趣點的句子，並以自動標記作為基礎，準備訓練資料，再利用 CRF++訓練的辨識模型。由於中華黃頁上的興趣點名稱包含不只一般商家名稱，也包括像是註冊人名、食物名或是類別名稱的興趣點，所以需要進行興趣點的篩選，以獲取更好的訓練資料。從實驗結果中可以看出，經過篩選後訓練的模型效能來到 0.788，從 0.665 大幅提升 15.6%。而利用人名辨識模型去除註冊人名的方法比正規表達式來的更好。此外，去除不包含興趣點的方法可以節省 37.3%的訓練時間，其效能和保留所有句子的方式不相上下。而去除不包含實體以及標籤之過短的正例也能提升效能。隨著資料量的增加，短實體的效能獲得改善，整體 F1 值也從一個搜尋結果的 0.788 提升到十個搜尋結果的 0.816，因此我們相信，若有足夠的硬體設備，足以負荷更多的資料，就能夠訓練出更好的模型。

在地址與興趣點關聯預測部分，我們提出三種不同準備反例的方法：隨機分配興趣點給地址、從地址的搜尋結果擷取之錯誤配對興趣點以及混合前兩者，其精準率分別為 0.746、

0.744 和 0.754，差異並不顯著。

整體系統方面，我們採取六種不同的組合對一百個新的地址進行測試，從結果中可以看出，當關聯分類模型能成功分類出正確的配對時，搭配上越好的興趣點辨識模型能夠提高效能，減少辨識錯誤興趣點的機率並大幅降低實驗時間，同時也提升整體系統效率。最後我們的系統能從搜尋結果中辨識地址，透過辨識出該地址附近的興趣點，再利用關聯分類模型配對地址與興趣點，找到該地址最有可能的興趣點，達到自動擴充以及自動挖掘的功能。其效率達到每個 IP 每天能爬取約 49 個新的興趣點。

在未來研究上，首先我們的興趣點實體篩選只保留括號，其餘符號以及英文數字皆被去除，然而實際上有許多興趣點是中英混雜或是夾帶數字，未來若能訓練出跨語言的辨識模型，就能辨識出更多興趣點。其次雖然三種不同的關聯分類模型效能差異並不大，但在系統測試時卻有非常大的不同，從學習曲線中可以看出，訓練資料五百時準確率即達到 0.739，若能手動標記從地址的搜尋結果中辨識非正例配對的興趣點，將原本被視為反例的興趣點修正成正例(例如公司縮寫或是連鎖店)，或許能提升模型效能，在系統面時亦能更符合實際狀況，找到更多更正確的候選人。此外，利用 Google 搜尋引擎會有次數的限制，因此一般我們需要使用多個 IP 或是浮動 IP 來提高系統效率。

參考文獻

[1] Chien-Lung Chou, Chia-Hui Chang, and Ya-Yun Huang. 2016. Boosted Web Named Entity Recognition via Tri-Training. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 16, 2, Article 10 (Oct. 2016), 23 pages.

[2] Chuang, Hsiu-Min, and Chia-Hui Chang. "Verification of poi and location pairs via weakly labeled web data." Proceedings of the 24th International Conference on World Wide Web. ACM, 2015.

[3] Bach, Nguyen, and Sameer Badaskar. "A review of relation extraction." Literature review for Language and Statistics II (2007).

[4] Kambhatla, Nanda. "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations." Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2004.

[5] Zhao, Shubin, and Ralph Grishman. "Extracting relations with integrated information

using kernel methods." Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2005.

[6] Brin, Sergey. "Extracting patterns and relations from the world wide web." International Workshop on The World Wide Web and Databases. Springer, Berlin, Heidelberg, 1998.

[7] Banko, Michele, et al. "Open Information Extraction from the Web." IJCAI. Vol. 7. 2007.

[8] Etzioni, Oren, et al. "Unsupervised named-entity extraction from the web: An experimental study." Artificial intelligence 165.1 (2005): 91-134.

[9] Agichtein, Eugene, and Luis Gravano. "Snowball: Extracting relations from large plain-text collections." Proceedings of the fifth ACM conference on Digital libraries. ACM, 2000.

[10] Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. "An algorithm that learns what's in a name." Machine learning 34.1 (1999): 211-231.

[11] McCallum, Andrew, and Wei Li. "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons." Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003.

[12] 黃雅筠, 張嘉惠, 周建龍. 基於已知名稱搜尋結果的網路實體辨識模型建立工具, ROCLING XXVII (2015).

[13] 高霆耀; 莊秀敏; 張嘉惠. 基於 Web 之商家景點擷取與資料庫建置. ROCLING XXVII (2015), 2015, 180.